Homework assignment

Use the template source file (`hw3.wl` in your repository), implement the denotational semantics described bellow.

Grammar rules

$$\begin{aligned}
program &::= compose[statement, program]|\\
&\quad\ \epsilon\\
statement &::= assign[varName, expression]|\\
&\quad\ while[b, c]\\
expression &::= number|\\
&\quad\ boolean|\\
&\quad\ plus[expr, expr]|\\
&\quad\ less[expr, expr]|\\
&\quad\ equal[expr, expr]|\\
&\quad\ or[expr, expr]|\\
&\quad\ and[expr, expr]|\\
&\quad\ value[varName]
\end{aligned} \tag{1}$$

Denotational Semantics

Value rules:

$$[\![n]\!] = \lambda env.n \tag{2}$$
$$[\![true]\!] = \lambda env.True \tag{3}$$
$$[\![false]\!] = \lambda env.False \tag{4}$$
$$[\![varName]\!] = varName \tag{5}$$

Statement rules:

$$[\![assign[varName, e]]\!] = [\![assign]\!]([\![varName]\!], [\![e]\!]) \tag{6}$$
$$[\![assign]\!] = \lambda v, e.\lambda env.env[v \mapsto e(env)] \tag{7}$$
$$[\![compose[stm, prg]]\!] = [\![compose]\!]([\![stm]\!], [\![prg]\!]) \tag{8}$$
$$[\![compose]\!] = \lambda s, p.\lambda env.p(s(env)) \tag{9}$$
$$[\![epsilon]\!] = \lambda env.env \tag{10}$$
$$[\![while[b, c]]\!] = [\![while]\!]([\![b]\!])([\![c]\!]) \tag{11}$$
$$[\![while]\!] = Y\,\lambda r.\lambda b.\lambda c.\lambda env.(b(env)?r(b)(c)(c(env)) : env) \tag{12}$$

Expression rules:

$$[\![value[varName]]\!] = [\![value]\!]([\![varName]\!]) \tag{13}$$
$$[\![value]\!] = \lambda v.\lambda env.env[v] \tag{14}$$
$$[\![binop[e_1, e_2]]\!] = [\![binop]\!]([\![e_1]\!], [\![e_2]\!]) \tag{15}$$
$$[\![plus]\!] = \lambda e_1, e_2.\lambda env.e_1(env) + e_2(env) \tag{16}$$
$$[\![less]\!] = \lambda e_1, e_2.\lambda env.e_1(env) < e_2(env) \tag{17}$$
$$[\![equal]\!] = \lambda e_1, e_2.\lambda env.e_1(env) = e_2(env) \tag{18}$$
$$[\![or]\!] = \lambda e_1, e_2.\lambda env.e_1(env)\ ||\ e_2(env) \tag{19}$$
$$[\![and]\!] = \lambda e_1, e_2.\lambda env.e_1(env)\ \&\&\ e_2(env) \tag{20}$$